

IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1-92. (Canceled)

93. (Previously Presented) A method for delivering byte code comprising:

- extracting from byte code a header, said header having class information descriptive of classes to be loaded;
- extracting a first time from said header;
- extracting a second time from said header;
- loading said classes after said first time and before said second time.

94. (Previously Presented) A method for handling byte code in a configurable manner, said method comprising:

- receiving a data stream;
- identifying a byte code header within the data stream; and
- extracting information from the byte code header, wherein the extracted information includes:
 - references to one or more classes, and
 - timing information corresponding the one or more classes, wherein the timing information specifies a first deadline; and
- completing the loading of the one or more classes before the first deadline.

95. (Previously Presented) The method of claim 94, wherein the timing information includes a second deadline, and wherein the method includes refraining from beginning the loading of the one or more classes before the second deadline.

96. (Previously Presented) The method of claim 94, wherein the timing information includes a second deadline, and wherein the method includes beginning the loading of the one or more classes after the second deadline.

97. (Previously Presented) The method of claim 96, wherein the first and second deadlines are specified as time stamps.

98. (Previously Presented) The method of claim 94, further comprising executing the byte code.

99. (Previously Presented) The method of claim 94, further comprising:
detecting whether said loading is not completed before the first deadline; and
reporting an error in response.

100. (Previously Presented) The method of claim 94, wherein the extracted information specifies the format of the byte code.

101. (Previously Presented) The method of claim 100, further comprising configuring the receiving of the data stream by using the format to identify additional byte code headers within the data stream.

102. (Previously Presented) The method of claim 94, wherein the extracted information specifies a delivery method for the byte code.

103. (Previously Presented) The method of claim 102, further comprising configuring the receiving of the data stream according to the delivery method.

104. (Previously Presented) The method of claim 94, wherein the extracted information also specifies the interactions of the byte code.

105. (Previously Presented) The method of claim 94, wherein the extracted information also specifies the behavior of the byte code.

106. (Previously Presented) A programmable control system for handling byte code transport, wherein the programmable control system comprises:

- a computer-readable medium having computer program code embodied therein,
 - wherein the computer program code is configured to cause the programmable control system to:

- receive a data stream including a byte code with a header;

- extract information from the header, wherein the information includes

- configuration information for the programmable control system, wherein said configuration information includes timing information to control the loading of classes; and

- time the loading of additional code or data relative to the timing information from the header.

107. (Previously Presented) The programmable control system of claim 106, wherein the information includes information descriptive of application program interfaces to be loaded.

108. (Previously Presented) The programmable control system of claim 107, wherein the application program interfaces to be loaded include a scene graphs processor application program interface.

109. (Previously Presented) The programmable control system of claim 107, wherein the application program interfaces to be loaded include a data stream decoder application program interface.

110. (Previously Presented) The programmable control system of claim 109, wherein the data stream decoder program interface uses the MPEG-1, MPEG-2, or MPEG-4 standard.

111. (Previously Presented) The programmable control system of claim 107, wherein the application program interfaces to be loaded include a network interface application program interface.

112. (Previously Presented) The programmable control system of claim 107, wherein the information includes a transport mechanism specifying either RTP (Real-time Transport Protocol) or MPEG-2.

113. (Previously Presented) A method for handling byte code transport comprising:
determining a byte code ready for transport;
constructing a header for the byte code; and
sending the header and the byte code attached as a data stream, wherein the header includes configuration and timing information for programmably receiving the byte code in a timely fashion.

114. (Previously Presented) The method of claim 113, wherein the sending is a broadcast to multiple destinations.

115. (Previously Presented) The method of claim 113, wherein the method is implemented on a server, and wherein said server supports multiple users subscribing to said broadcast, multiple users can subscribe to the server, and wherein the server multicasts the byte code and the header to the multiple users.

116. (Previously Presented) The method of claim 113, wherein multiple users can request make requests affecting the determination of the byte code.

117. (Previously Presented) The method of claim 113, wherein protocol for the sending of the data stream is UDP (User Datagram Protocol).

118. (Previously Presented) A method for receiving byte code, the method comprising:

- receiving a data stream having one or more packets;
- reading a byte code header from the data stream; and
- extracting information from the byte code header, wherein the extracted information includes:
 - a first timestamp specifying a first time by which a first class will be loaded,
 - a second timestamp specifying a second time before which loading of the first class will not be started, and
 - information identifying one or more additional class to be loaded, wherein the first class is dependent upon the one or more additional classes.

119. (Previously Presented) A method for receiving byte code, the method comprising:

- receiving a data stream having one or more packets;
- reading a byte code header from the data stream; and
- extracting information from the byte code header, wherein the extracted information includes:
 - a load-by timestamp specifying a load-by time by which a first class will be loaded,
 - a start loading timestamp specifying a start loading time before which loading of the first class will not be started, and
 - information identifying one or more additional classes to be loaded, wherein the first class is dependent upon the one or more additional classes.

120. (Previously Presented) The method of claim 119, wherein the data stream includes start loading timestamps and load-by timestamps for the one or more additional classes, wherein the start loading time of the first class is later than the load-by times of each of the additional classes upon which the first class depends.

121. (Previously Presented) The method of claim 119, wherein the start loading time of the first class is later than the load-by times of each of the additional classes upon which the first class depends.

122. (Previously Presented) A method for receiving byte code, the method comprising:

- receiving a data stream having one or more packets;
- reading a byte code header from the data stream; and
- extracting information from the byte code header, wherein the extracted information includes:
 - a first timestamp specifying a first time before which a first class will be loaded,
 - a second timestamp specifying a second time before which loading of the first class will not be started, and
 - information identifying a second class to be loaded, wherein the first class is dependent upon the second class.

123. (Previously Presented) The method of claim 122, further comprising:
extracting information from the data stream including a third timestamp specifying a third time by which the second class will be loaded, wherein the third time is before the second time.

124. (Previously Presented) The method of claim 122, wherein the data stream includes a payload associated with the byte code header, wherein the payload and byte code header are packetized within the data stream.

125. (Previously Presented) The method of claim 124, wherein the first class is part of the payload.

126. (Previously Presented) The method of claim 124, wherein the first class and the second class are part of the payload.

127. (Previously Presented) The method of claim 124, wherein the first class and the second class are in different payloads.

128. (Previously Presented) A method for receiving byte code, the method comprising:

- receiving a data stream having two or more classes, wherein each class has a header, wherein the headers and classes are packetized into one or more packets;
- reading a byte code header from the data stream, wherein the byte code header is distributed amongst one or more of the packets; and
- extracting information from the byte code header, wherein the extracted information includes:
 - a first timestamp specifying a first time before which a first class will be loaded,
 - a second timestamp specifying a second time before which loading of the first class will not be started, and
 - information identifying a second class to be loaded, wherein the first class is dependent upon a second class.

129. (Previously Presented) The method of claim 128, wherein the data stream includes a second header distributed amongst one or more of the packets, wherein the method further includes:

- reading the second header from the data stream, and
- extracting information from the second header, wherein the extracted information from the second header includes:
 - a third timestamp specifying a third time before which the second class will be loaded, and

a fourth timestamp specifying a fourth time before which loading of the second class will not be started, wherein the second time is before the third time.

130. (Currently Amended) The method of claim 128, wherein the byte code is [[Java]] JAVA byte code, and wherein the first class and second class are [[Java]] JAVA classes.

131. (Previously Presented) The method of claim 128, wherein the two or more classes are compressed, wherein the method further comprises decompressing the first class and the second class.

132. (Previously Presented) A method comprising:
receiving a data stream;
identifying a byte code header within the data stream;
extracting information from the byte code header, wherein the extracted information includes:
a first reference to a first class,
a second reference to a second set of one or more classes upon which the first class depends on, and
timing information corresponding the first class, wherein the timing information specifies a first time and a second time;
loading the second set of one or more classes;
beginning loading the first class no sooner than the first time and after the second set of one or more classes has completed loading; and
completing the loading of the first class by the second timestamp, wherein the first class are dependent upon the second set of one or more classes.

133. (Previously Presented) A method comprising:
receiving a data stream;
identifying a byte code header within the data stream;

extracting information from the byte code header, wherein the extracted information includes:
a first reference to a first set of one or more classes,
a second reference to a second set of one or more classes, and
timing information corresponding to the first set of one or more classes,
wherein the timing information specifies a first time and a second time;
loading the second set of one or more classes;
beginning loading of the first set of one or more classes no sooner than the first time and after the second set of one or more classes has completed loading; and
completing the loading of the first set of one or more classes by the second time,
wherein the first set of classes are dependent upon the second set of one or more classes.

134. (Previously Presented) The method of claim 133, further comprising automatically executing the first set of one or more classes once the first set of one or more classes has completed loading.

135. (Previously Presented) The method of claim 133, further comprising asserting an error condition if the first set of one or more classes has not completed loading by the second time.

136. (Previously Presented) The method of claim 133, further comprising asserting an error condition if the second set of one or more classes has not completed loading by the first time.

137. (Currently Amended) The method of claim 133, wherein the byte code is JAVA byte code.

138. (Previously Presented) The method of claim 133, wherein the first set of one or more classes and the byte code header are packetized into one or more packets.

139. (Previously Presented) The method of claim 138, wherein the second set of one or more classes is dependent on one or more additional classes.

140. (Previously Presented) The method of claim 138, wherein the second set of one or more classes and a second byte code header are packetized into one or more packets.

141. (Previously Presented) The method of claim 140, wherein the second byte code header includes:

a third timestamp specifying a third time before which the second set of one or more classes is loaded, and

a fourth timestamp specifying a fourth time before which loading of the second set of one or more classes will not be started.

142. (Previously Presented) The method of claim 141, wherein the third timestamp is a load-by timestamp, and wherein the fourth timestamp is a start loading timestamp.

143. (Currently Amended) The method of claim 133, wherein the byte code is `[[Java]] JAVA` byte code, and wherein the first and second sets of one or more classes are sets of one or more `[[Java]] JAVA` classes.

144. (Previously Presented) A computer program embodied on a computer-readable medium, wherein the computer-program is executable to:

(a) receive a data stream having one or more packets;

(b) read a byte code header in the data stream;

(c) extract information from the byte code header, wherein the extracted information includes:

a start loading timestamp specifying a start loading time for a first class in the data stream,

a load-by timestamp specifying a load-by time for the first class in the data stream, and

information identifying a second class in the data stream, wherein the first class is dependent upon the second class;

- (d) start loading the first class no sooner than the start loading time;
- (e) complete loading of the first class by the load-by time;
- (f) determine whether the first class completed loading by the second time; and
- (g) assert an error condition if the first class did not complete loading by the load-by time.

145. (Currently Amended) The computer program of claim 144, wherein the byte code is [[Java]] JAVA byte code, wherein the first class and the second class are [[Java]] JAVA classes.

146. (Previously Presented) The computer program of claim 144, further configured to execute the first class after the first class has completed loading.

147. (Previously Presented) The computer program of claim 144, further configured to execute the first class after the first class has completed loading.

148. (Previously Presented) A method for preparing byte code for transmission, the method comprising:

creating a data stream having one or more payloads, wherein a first of the one or more payloads includes a first class, and

adding a byte code header to a first payload in the data stream, wherein the byte code header includes:

a first timestamp specifying a first time by which a first class will be loaded,

a second timestamp specifying a second time after which loading of the first class will start, and

information identifying at least a second class to be loaded, wherein the first class is dependent upon at least the second class.

149. (Previously Presented) The method of claim 148, wherein the second class is in the first payload.
150. (Previously Presented) The method of claim 148, wherein the second class is in a second payload.
151. (Previously Presented) The method of claim 150, further comprising:
adding a second byte code header to the second payload, wherein the second byte code header includes:
a third timestamp specifying a third time by which the second class will be loaded, and
a fourth timestamp specifying a fourth time after which loading of the second class will start, wherein the third time is before the second time.
152. (Previously Presented) The method of claim 151, wherein the second byte code header further includes information identifying a third class to be loaded, wherein the second class is dependent upon the third class.
153. (Previously Presented) The method of claim 148, further comprising packetizing the headers and payloads into one or more packets.
154. (Previously Presented) The method of claim 148, wherein each payload has a single class.
155. (Previously Presented) The method of claim 148, further comprising compressing the first class and the second class.
156. (Previously Presented) A method for preparing byte code for transmission, the method comprising:

creating a stream of data having one or more payloads, wherein a first of the one or more payloads includes a first class and a second class; and
adding a byte code header to a first payload in the data stream, wherein the byte code header includes:
information specifying a load-by time by which a first class in the data stream will be loaded,
information specifying a start loading time for the first class, and
information identifying a second class to be loaded before the first class, wherein the first class is dependent upon the second class.

157. (Previously Presented) The method of claim 156, further comprising packetizing the headers and payloads into one or more packets.

158. (Previously Presented) The method of claim 156, wherein each payload has one or more classes.

159. (Previously Presented) The method of claim 156, further comprising compressing the first class and the second class.

160. (Previously Presented) A computer program embodied on a computer-readable medium, wherein the computer program is executable to prepare byte code for transmission by:

creating a data stream having one or more packets, wherein a first of the one or more packets has a first class, and wherein a second of the one or more packets has a second class; and
adding a byte code header to a first packet in the data stream, wherein the byte code header includes:
a first timestamp specifying a first time by which a first class will be loaded,
a second timestamp specifying a second time after which loading of the first class will start, and

information identifying at least a second class to be loaded, wherein the first class is dependent upon the second class.

161. (Previously Presented) A method for handling byte code in a configurable manner, said method comprising:
- receiving a data stream;
 - identifying a byte code header within the data stream; and
 - extracting information from the byte code header, wherein the extracted information includes:
 - references to one or more classes, and
 - timing information corresponding the one or more classes, wherein the timing information specifies a first deadline and a second deadline;
 - starting the loading of the one or more classes not before the first deadline, and
 - completing the loading of the one or more classes not later than the second deadline.
162. (Previously Presented) The method of claim 161, further comprising:
- detecting whether the loading is not completed before the first deadline; and
 - reporting an error in response.
163. (Previously Presented) The method of claim 161, wherein the byte code header also includes a reference to a set of dependent classes that must be loaded before the one or more classes can be loaded.
164. (Previously Presented) The method of claim 163, further comprising detecting that one or more of the classes are executable and automatically executing the executable classes after successful loading.
165. (Previously Presented) The method of claim 164, wherein the first and second deadlines are specified as time stamps.

166. (Previously Presented) The method of claim 161, wherein the extracted information specifies the format of the byte code.
167. (Previously Presented) The method of claim 166, further comprising configuring the receiving of the data stream by using the format to identify additional byte code headers within the data stream.
168. (Previously Presented) The method of claim 161, wherein the extracted information specifies a delivery method for the byte code.
169. (Previously Presented) The method of claim 168, further comprising configuring the receiving of the data stream according to the delivery method.
170. (Previously Presented) The method of claim 161, wherein the extracted information also specifies the interactions of the byte code.
171. (Previously Presented) The method of claim 161, wherein the extracted information also specifies the behavior of the byte code.
172. (Previously Presented) A programmable control system for handling byte code transport, wherein the programmable control system comprises:
a computer-readable medium having computer program code embodied therein,
wherein the computer program code is configured to cause the
programmable control system to:
receive a data stream including a byte code with a header;
extract information from the header, wherein the information includes
configuration information for the programmable control system, wherein
said configuration information includes timing information to control the
loading of classes; and
time the loading of additional code or data relative to the timing information from
the header.

173. (Previously Presented) The programmable control system of claim 172, wherein the information includes information descriptive of application program interfaces to be loaded.

174. (Previously Presented) The programmable control system of claim 173, wherein the application program interfaces to be loaded include a scene graphs processor application program interface.

175. (Previously Presented) The programmable control system of claim 173, wherein the application program interfaces to be loaded includes a data stream decoder application program interface.

176. (Previously Presented) The programmable control system of claim 175, wherein the data stream decoder program interface uses the MPEG-1, MPEG-2, or MPEG-4 standard.

177. (Previously Presented) The programmable control system of claim 173, wherein the application program interfaces to be loaded includes a network interface application program interface.

178. (Previously Presented) The programmable control system of claim 173, wherein the information includes a transport mechanism specifying either RTP (Real-time Transport Protocol) or MPEG-2.

179. (Previously Presented) A method for handling byte code transport comprising:
determining a byte code ready for transport;
constructing a header for the byte code; and
sending the header and the byte code attached as a data stream, wherein the
header includes configuration and timing information for programmably
receiving and loading the byte code in a timely fashion.

180. (Previously Presented) The method of claim 179, further comprising automatically starting the execution of the byte code.

181. (Previously Presented) The method of claim 179, wherein the sending is performed in a multicast broadcast fashion to multiple recipients.

182. (Previously Presented) The method of claim 181, wherein the method is implemented on a server, and wherein said server supports multiple users subscribing to said broadcast, multiple users can subscribe to the server, and wherein the server multicasts the byte code and the header to the multiple users.

183. (Previously Presented) The method of claim 179, wherein multiple users can request make requests affecting the determination of the byte code.

184. (Previously Presented) The method of claim 179, wherein protocol for the sending of the data stream is UDP (User Datagram Protocol).